



Videoconferencing over OpenFlow Networks: An Optimization Framework for QoS Routing

Djamel-Eddine Henni, Abdelghani Ghomari, Yassine Hadjadj-Aoul

► To cite this version:

Djamel-Eddine Henni, Abdelghani Ghomari, Yassine Hadjadj-Aoul. Videoconferencing over Open-Flow Networks: An Optimization Framework for QoS Routing. 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Oct 2015, LIVERPOOL, United Kingdom. pp.491-496. hal-02266447

HAL Id: hal-02266447

<https://inria.hal.science/hal-02266447>

Submitted on 14 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Videoconferencing Over OpenFlow Networks: An Optimization Framework for QoS Routing

Djamel-Eddine HENNI
University of Oran Es-Senia
Computer Science Department
RIIR Laboratory, Oran, Algeria
Email: henni_djamel@yahoo.fr

Abdelghani GHOMARI
University of Oran Es-Senia
Computer Science Department
RIIR Laboratory, Oran, Algeria
Email: ghomari65@yahoo.fr

Yassine HADJADJ-AOUL
University of Renne 1,
Rennes, France
IRISA Laboratory
Email: yhadjadj@irisa.fr

Abstract—The recent emergence of OpenFlow, in the field of networks' control, marks a major evolution in networking. Indeed, Openflow comes up with a centralized control plane (real or virtual) and a remote control of the data plan, which enable network's evolution with unprecedented rapidity. This evolution, come, however, with new challenges and open issues. The data path portion still resides on networking devices, while high-level routing decisions are moved to a separate remote server, which may impact the QoS of the supported services. Particularly, there is a need to guarantee at least the QoS that was formally provided using older and classical distributed protocols. This paper focuses on an improvement of the traditional OpenFlow controllers by providing QoS Routing support for Videoconferencing over OpenFlow networks. The simulation results under Mininet clearly demonstrate the effectiveness of the proposed approach.

Index Terms—SDN; QoS; Routing; Visioconference;

I. INTRODUCTION

The forecasted dramatic growth of the number of connected devices and the volume of traffic originating particularly from multimedia application, such as videoconferencing and video streaming, calls for shifting from the current distributed management of the network to a centralized control¹. Indeed, the network control centralization provides an efficient and attractive way to tackle the challenge of increased management complexity.

OpenFlow is certainly the more significant and popular open standard, which is initially proposed in [3] with the objective to centralize (even partially) the management and the control of the network's nodes. It is, indeed, the most widely deployed Software Defined Networking (SDN) technology. The main idea behind SDN consists in decoupling the control and data plans of a network. A software-based controller is, thus, responsible for managing the forwarding information of one or more network devices that handle the forwarding of traffic according to the rules set by the controller.

Delivering services with guaranteed quality of service (QoS) has been historically considered as one of the main issues for network operators (NO). Nowadays, MPLS represents the most advanced mechanism for QoS support [8], which provides an ultra-fast switching capability. However, MPLS

lacks real-time re-configuration and adaptation to network conditions. In the context of OpenFlow, QoS is at its infancy. Indeed, the OpenFlow 1.0 specifications only allow setting the network type of service for a particular flow by adding rules to enqueue its packets to a specific queue on a specific port. Even if the OpenFlow 1.3 specifications present more features, there is no models or architectures in terms of QoS deployment that has been implemented in the current Internet design.

In this paper, we propose a new controller design that enables QoS routing for videoconferencing flow delivery over OpenFlow networks in order to achieve the required QoS metrics. We propose a dynamical QoS routing for QoS-enabled flows (i.e. videoconferencing traffic) which focus on protecting such traffic over non constrained flows (i.e. non prioritized). The dynamic QoS routing problem is formulated as a Constrained Shortest Path (CSP) problem, which is solved using the Delay-Constrained Least-Cost (DCLC) algorithm.

The rest of the paper is organized as follows: section II discusses the related works. Section III proposes an Openflow Based QoS Routing Architecture and the problem formulation. Simulation settings and results are discussed in section IV. Section V draws some conclusions and future work.

II. RELATED WORKS

Routing, generally, consists in two basic tasks or functions. The first one is to collect the state information and keep it up to date. The second task is to find a feasible path for a new connection based on the collected information. Thus, the performance of any routing algorithm depends, directly, on how well these two tasks were conducted.

QoS routing has been a major concern of both academia and network operators for many years. In fact, a very huge number of studies have been made in such topic to improve the users' quality of experience (QoE). In [6], the authors provide an overview of QoS routing techniques over next generation high-speed networks. They present different QoS routing problems, their challenges, the QoS routing strategies and a comparison of existing routing algorithms. Early, resources' reservation was considered as an efficient mean to protect QoS-constrained connections. In [7], [9] the authors have considered resource reservation (bandwidth, CPU time, buffer...etc.) at the connections' establishment to isolate concurrent connections sharing

¹It is meant by centralization, a logical or a physical centralization of the control plan.

common links. In [19] a new routing algorithm, named the Wang-Crowcroft algorithm, was proposed to find a bandwidth-delay-constrained path using Dijkstra shortest-path algorithm. At a first step of this algorithm, all the links with a bandwidth less than the requirement are eliminated, which allows any paths in the resulting graph to satisfy the constraint in terms of bandwidth. Then the shortest path in terms of delay is selected. In [18] the authors proposed a novel connectionless QoS routing framework, where QoS constraints are carried by every packet and in which the routing decisions are made hop by hop at the packet level improving the scalability of the network. In [10] the authors treat various approximation algorithms for MCP (multi constrained path) problems and propose an improved approximation scheme. A recent work of the authors, in [12], presented an optimization framework for the OpenFlow controller in order to provide QoS support for SVC (Scalable Video Coding) over an OpenFlow network where they provide lossless-QoS to the base layer of the SVC encoded video and a lossy-QoS or a best-effort to the enhancement layers of SVC video.

III. AN OPENFLOW-BASED QoS ROUTING ARCHITECTURE

A. Controller Design and QoS Routing Support in OpenFlow

SDN is characterized by a centralized control plane, which allows moving part of the decision-making logic of network devices to external controllers [15]. This characteristic provides the controller device with the ability to have an overall view of the network infrastructure, thus becoming aware of network elements and network characteristics. SDN architecture can be viewed as having three following layers: the infrastructure layer, the control layer and the applications layer. These layers are illustrated in Fig 1. The Openflow protocol is playing a key role in such architecture by providing a link and an interface between the SDN controllers using a network Operating System (OS) and the interconnection devices (i.e. the infrastructure elements).

As in the classical SDN-based architectures, the routing tasks, in the proposed architecture, are, also, performed at the control layer in which the controllers have a global abstracted view of the network topology. In fact, as mentioned above, the existing network architectures are not really suited with the critical applications. Indeed, when a packet arrives at a router, it checks the packet's source and destination address pair with the entries of the routing table, and forwards it according to predefined rules configured by the network operator. The springiness of Openflow allow us, however, to remedy this weakness by defining different routing rules associated with the data flows as mentioned later. The controller is the brain of the network, it runs customized programs to decide which rules and actions are going to be installed to control packet forwarding in each switch element through interfaces with these forwarders, which effectively apply the rules.

B. Defining Flows in Openflow Networks

In OpenFlow, we can define flows using different methods. Flows can contain similar or different types of packets. For

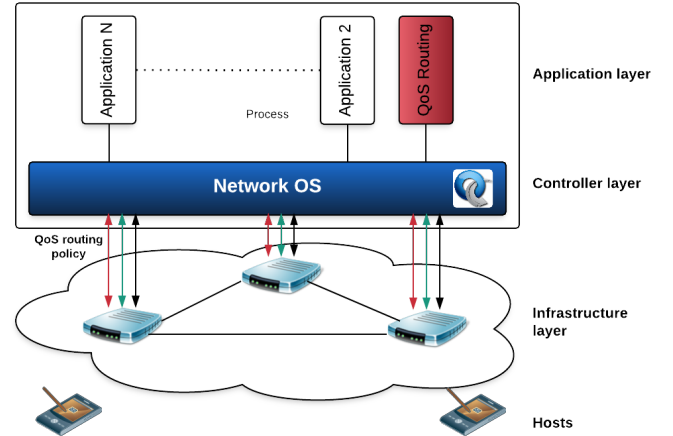


Fig. 1. SDN architecture

example, packets with the TCP port number 21 (reserved for FTP) can be a flow definition, or packets having RTP header may indicate a flow which carries voice, video or both. Flows can be set as a combination of headers as illustrated in Fig.2. In Openflow-based networks, the devices store the flows and their associated properties in flow tables, which are processed as a pipeline [4].

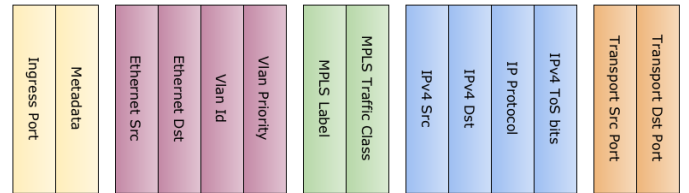


Fig. 2. Identification of flow fields in Openflow

In order to differentiate between flows, we exploit the Openflow's flow-based forwarding paradigm, where flows may be determined using the following packet header fields or values [12]:

- TOS (Type of Service) field of IPv4.
- Traffic class header field in MPLS.
- Traffic class field in IPv6.
- IP addressing if the source IP address of the flow is known.
- Transport source and/or destination port numbers.

C. Optimization of QoS routing using an Openflow-based controller

In order to implement a QoS mechanism, we must define the QoS metrics that will be used, while defining the application's requirement in terms of QoS metrics.

Videoconferencing enables real time communication by allowing people at two or more sites to communicate with each other. In videoconferencing, in addition to voice-channels, as in the conventional public switched telephony network (PSTN), video channels are also exchanged from the various sites [13]. Each site has one or more cameras, microphones, loudspeakers and monitors, as well as a list of supported codecs. This type of conferencing aims to create an effect called Virtual Presence. The QoS metrics requirements for videoconferencing are given in Table I [17].

Response time (ms)	delay (ms)	Jitter (ms)	Loss Rate	Error Rate
< 100	< 200	< 400	< 0.01%	< 0.01%

TABLE I
VIDEOCONFERENCING QoS METRICS REQUIREMENTS

In the following, we formulate the dynamic QoS routing problem as a Delay-Constrained Least-Cost (DCLC). In fact, as explained above, videoconferencing is an interactive multi-media application which requires a strict end-to-end delay and packets loss requirement.

In our proposal, a network is formulated as a direct network graph $G(N, L)$ where N is the set of nodes and L the set of all the links, so that link (i, j) is an ordered pair, which is outgoing from node i and ingoing to node j . We consider that R_{xy} is the set of all possible routes from source x to destination y . For any route $r \in R_{xy}$ we define the cost function $f_C(r)$ and the delay function $f_D(r)$, which are calculated as follows:

$$f_C(r) = \sum_{(i,j) \in r} c_{ij}, \quad f_D(r) = \sum_{(i,j) \in r} d_{ij} \quad (1)$$

where the coefficients c_{ij} and d_{ij} represent, respectively, the cost in capacity and delay for the link (i, j) . The best path is the one minimizing the cost function $f_C(r)$ subject to the delay function $f_D(r)$, which must be less than or equal to a specified value D_{max} .

We considered cost metric is calculated as the weighted sum of packet loss measure and delay as follows:

$$c_{ij} = (1 - \alpha)d_{ij} + \alpha p_{ij} \text{ for } 0 \leq \alpha \leq 1, \forall (i, j) \in L \quad (2)$$

where p_{ij} represents the packet loss measure for the QoS constrained traffic and best-effort flows on link (i, j) , and α is the scale factor which determines the relative importance of the delay and the packet losses depending on network and traffic importance. The route selection would be more sensitive to packet losses on the QoS route for large network and more sensitive to delay for a small one. The formula for p_{ij} is defined as below:

$$p_{ij} = \begin{cases} \frac{Q_{ij} + T_{ij} - B_{ij}}{Q_{ij} + T_{ij}}, & B_{ij} < Q_{ij} + T_{ij} \\ 0, & B_{ij} \geq Q_{ij} + T_{ij} \end{cases} \quad (3)$$

where B_{ij} is the bandwidth of the link (i, j) , Q_{ij} and T_{ij} are the amounts of QoS-constrained traffic and best-effort traffic observed on the link (i, j) , respectively. The route management function of the controller collects data from forwarders (p_{ij}, d_{ij}) and passes them to the route calculation. At the forwarder layer, the considered parameters are estimated as follows:

- Packet loss measure (p_{ij}) is calculated using (3) where B_{ij} , Q_{ij} and T_{ij} are required parameters for calculation. Monitoring of the per-flow traffic amounts (Q_{ij} and T_{ij}) on each link is done using the Openflow protocol which maintains the per-flow counters in the forwarders [4].
- Delay measure can be obtained by averaging the observed delay using time stamping as in RTP [11].

The problem proposed in this work is an example of minimum cost flow optimization problems in network optimization theory. In the literature, there are many algorithms available in order to solve min-cost flow problems. We solved the proposed problem by using the algorithm discussed in [16] and illustrated in Algorithm 1.

When the route management function updates the QoS indicating parameters or the topology management function detects a topology change, the route calculation function runs the DCLC algorithm to solve the CSP problem (finding the best path which minimize the cost function $f_C(r)$ subject to the delay function $f_D(r)$, which must be less than or equal to a specified value D_{max}). Then, the controller updates the forwarders flow tables accordingly and the QoS routes are dynamically set for videoconferencing traffic.

D. The DCLC Algorithm

Step 0: Start.

Step 1: If there is more than one path with minimum cost, we find the least cost path between s (source) and d (destination) with minimum delay, and let this path named q.

Step 2: Check the delay of the selected path in (step1) is less than or equal the delay constrained.

Step 3: If the result of (step 2) is true then the suitable path is q.

Step 4: If the result of step 2 is false then let p be the shortest path in corresponding to delay (i.e. the path that has minimum delay).

Step 5: check if the delay of the path p is greater than delay constrained.

Step 6: If the result of (step 5) is true then there is no feasible path (return NULL).

Step 7: Create a flag variable named continue with Boolean type having initial value = True, Continue= True.

Step 8: Building a close loop with a condition continue=true and exit the loop just when continue =false. While continue

do

Step 9: If the result of step 5 is false then calculate the parameter of constructing the mixed weight (α), where: $\alpha = \frac{Cd-d(p)}{c(p)-c(q)}$. Cd is the delay constrained, d(p) is the delay of the path, c(p) is the cost of the path and c(q) is the cost of the path q.

Step 10: Convert each path to one parameter (w), where $w = \text{delay} + \alpha * \text{cost}$ for each path, Then let r is the minimum path corresponding to (w) value.

Step 11: Check: If cost of the path(r) equal the cost of path(q) or the cost of the path p. If $(c(r) = c(q) \text{ or } c(r) = c(p))$ then

Step 12: If the result of the condition in (step 11) is true then let continue = false to exit loop (step 15), Continue=False

Step 13: If the result of the condition in (step 11) is false then (step 14) Else

Step 14: Let p = the minimum path(r).

Step 15: p is the algorithm choice of most suitable path. Return p .

Step 16: End.

Algorithm DCLC (G,s,t,c,d,Cd)

```

q ← ModDijk(c, d)
if d(q) ≤ Cd then
  | return q
else
  | p ← Dijk(d)
end
if d(p) > Cd then
  | return Null
else
  | Continue=TRUE
end
while Continue=TRUE do
   $\alpha \leftarrow \frac{Cd-d(p)}{c(p)-c(q)}$ 
  w=delay+ $\alpha$ *cost for each path
  r ← Dijk(d +  $\alpha$ c);
  if (c(r) = c(q)) or (c(r) = c(p)) then
    | Continue=False
    | Return p
  else
    | p ← r
  end
end

```

Algorithm 1: The DCLC Algorithm

IV. SIMULATION AND RESULTS DISCUSSION

Having described the details of our proposal, we now direct our attention on evaluating its performance in realistic use cases. We particularly focus, in the following, on studying the impact of the use of QoS controller on multimedia flows when the network is overloaded.

A. Experimental Environnement

In order to simulate the OpenFlow architecture and test the performances of the proposed routing formulations, we used the mininet emulator [5]. The proposed application

is integrated within the floodlight controllers [2]. A simple network topology, shown in Fig 3, was created to compare the efficiency of the proposed mechanism against the former one.

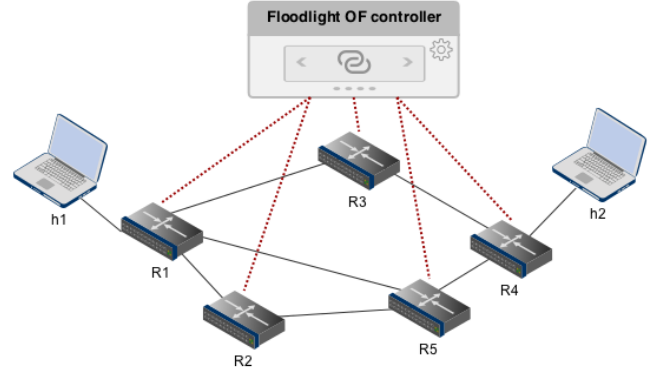


Fig. 3. The topology used in the simulations

The two hosts h1 and h2 are separated by five Open virtual switches and controlled by the floodlight controller. We have simulated three scenarios in which udp and tcp flows are generated between h1 and h2 at different rates in order to simulate different communication quality. Table II and Table III summarize the simulation parameters (udp rate, tcp rate, simulation time...etc).

Parametres	Values
Architecture	Openflow
Flows	TCP UDP
Simulation Time	240s
TCP Rate	800 Kbps
Bandwidth (R1-R5)	1Mbps
Bandwidth (R1-R2-R5)	2Mbps
Bandwidth (R1-R3-R4-R5)	3Mbps

TABLE II
SIMULATIONS PARAMETERS

Scenario	Udp Rate (Kbps)
First	384
Second	512
Third	784

TABLE III
UDP RATES

The udp flows represent a videoconferencing communication between h1 and h2 (QoS flow) while tcp flows are here in order to create a network congestion between h1 and h2 (best effort flow). Flows are generated using [1]. In the first scenario we generate udp flows at a rate of 384 Kbps between h1 and h2 without congesting the network with tcp flows and then we measure the packet loss, the jitter and the delay, then after, we congest the network with tcp flows without the use

of the QoS controller and we measure the same parameters (packet loss, jitter and delay). After that, we implement the QoS controller using eclipse then we run the simulation with the same parameters (udp flows at a rate of 384Kbps with presence of tcp flows). Table IV summarizes the results of this first scenario.

Simulation	TCP	Controller	loss (%)	Jitter (ms)	Delay (ms)
1	No	w/o QoS	0.00	5	20
2	Yes	w/o QoS	15.50	120	565
3	Yes	w/QoS	0.01	8	26

TABLE IV
OBTAINED RESULTS FOR AN UDP RATE AT 384 KBPS

In the second scenario, we take the same parameters of the simulation above but this time with an udp flow at a rate of 512 Kbps. The simulation results of this scenario are presented in Table v.

Simulation	TCP	Controller	loss (%)	Jitter (ms)	Delay (ms)
1	No	w/o QoS	0.00	6	20
2	Yes	w/o QoS	39.40	250	680
3	Yes	w/QoS	0.02	7	21

TABLE V
OBTAINED RESULTS FOR AN RDP RATE AT 512 KBPS

In the third scenario, we take the same parameters of the simulation above but this time with an udp flow at a rate of 784 Kbps. The simulation results of this scenario are presented in Table VI.

Simulation	TCP	Controller	loss (%)	Jitter (ms)	Delay (ms)
1	No	w/o QoS	0.00	6	21
2	Yes	w/o QoS	58.05	310	740
3	Yes	w/QoS	0.02	6	22

TABLE VI
OBTAINED RESULTS FOR AN UDP RATE AT 784 KBPS

Now, We simulate a real videoconference between h1 and h2 in two scenario using the same above topology where in the first scenario we use a video with 320x240 resolution with a frame frequency of 25 fps and coded with MPEG 2 codec. We generate a TCP flow between h1 and h2 in order to overload the network while they are in conference then we measure the QoE by calculation the PSNR for the received video with and without the application of the QoS mecanism. The obtained graph is shown in Fig 4.

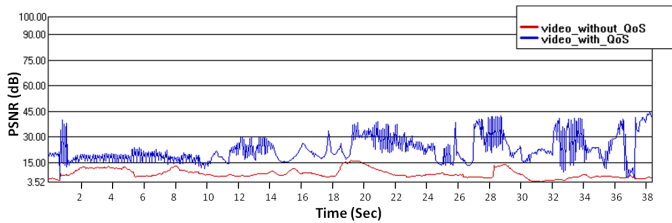


Fig. 4. Comparison between received video (320x240) with and without QoS

In the second scenario we change the video resolution to 352x288. The frame frequency and the used codec are the same as in the above scenario then we generate a TCP flow between h1 and h2 while they are in conference and we measure the QoE by calculation the PSNR for the received video with and without the application of the QoS mecanism. The obtained graph is shown in Fig 5.

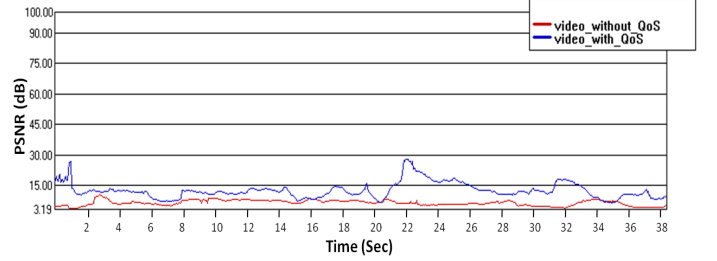


Fig. 5. Comparison between received video (352x288) with and without QoS

B. Results Discussion

When we generate udp flows without tcp, the network is less congested where the communication between h1 and h2 follows the shortest path R1-R5, in this case and whatever the used udp rate is, it is quite clear that we obtain an acceptable packet loss, jitter and delay since there is no congestion in the network. When we generate udp flows under a congested network without using a QoS controller, the udp communication (QoS flow) between h1 et h2 follows and share the shortest path R1-R5 with the best effort flow, therefore more packet loss occurs and we measure a significant delay and jitter due to the congested shortest path R1-R5. When we generate udp flows under a congested network with the use of QoS Controller, we obtain a significant reduction in terms of packet loss, jitter and delay as shown in the tables above whatever the quality of the udp communication between h1 and h2 which follows another path instead of the congested shortest path R1-R5.

V. CONCLUSIONS AND FUTURE WORK

QoS mechanisms allow network administrators to use existing resources efficiently and ensure the required level of service without the need of expanding or over provisioning their networks. In this paper, we advocate the use of QoS Routing in OpenFlow based networks and have showed through simulations that the shortest path is not always the best path to choose as it is done through classical networks. SDN and OpenFlow allow us to make less effort with more flexibility to pass new proposition and experimentation in terms of protocols and techniques. In this paper we have proposed a new controller design which takes into consideration path selection depending on criticality of the flow and provides a QoS routing for these applications. As part of our future work, we plan to study other case-studies to validate the broader applicability of our work on more larger networks.

REFERENCES

- [1] "DITG [Online]. Available: <http://traffic.comics.unina.it/software/itg/>.
- [2] "Floodlight [Online]. <http://www.projectfloodlight.org/floodlight/>.
- [3] "OpenFlow [Online]. Available: <http://openflowswitch.org>.
- [4] "OpenFlow Switch Specification Version 1.1.0, [Online] Available: <http://openflow.org>, 2011.
- [5] B. Lantz, B. Heller, and N. McKeown. "A network in a laptop: rapid prototyping for software-defined networks". In *Proceedings of the 9th ACM SIGCOMM workshop on Hot Topics in Networks*, ACM, 2010.
- [6] S. Chen and K. Nahrstedt. "An overview of quality of service routing for next-generation high-speed networks: Problems and solutions". *IEEE Netw.*, Vol. 12(6):pp. 64–79, Nov/Dec 1998.
- [7] D. Ferrari and D. C. Verma. "A Scheme for Real-Time Chonnet Establishment in Wide-Area Networks". *IEEE JSAC.*, Vol. 8(3):pp. 368–379, April 1990.
- [8] E. Rosen, A. Viswanathan and R. Callon. "Multiprotocol Label Switching Architecture" RFC 3031, January 2001.
- [9] L. Zhang et al. "RSVP": A New Resource Reservation Protocol. *IEEE Network.*, September 1993.
- [10] G. Xue and W. Zhang and J. Tang and K. Thulasiraman. "Polynomial time approximation algorithms for multi-constrained QoS routing". *IEEE ACM Trans. Netw.*, Vol. 16(3):pp. 656–669, June 2008.
- [11] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson,. "RTP: A Transport Protocol for Real-Time Applications", Internet Engineering Task Force, RFC 3550, Jul 2003.
- [12] Hilmi E. Egilmez, A. Murat Tekalp and S. Civanlar. "An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks". *IEEE TRANSACTIONS ON MULTIMEDIA.*, Vol. 15(3):pp. 710–715, April 2013.
- [13] J.J. Vleeschauwer, D.D. Buchli, M.J.C. and Kooij R.E. "Tuning an IP-based Network Transporting Telephony and Videophony", 2001.
- [14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner. "OpenFlow: Enabling Innovation in Campus Networks". *ACM SIGCOMM CCR*, Vol. 38(2):pp. 69–74, April 2008.
- [15] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao. "Are we ready for sdn? implementation challenges for software-defined networks". *Communications Magazine IEEE.*, Vol. 51(7):pp. 36–43, 2013.
- [16] Waleed A. Mahmoud and Dheyaa J. Kadhim. "A Proposal Algorithm to Solve Delay Constraint Least Cost Optimization Problem ". *Journal of Engineering*, (1):pp. 155–160, January 2013.
- [17] Y. Chen, T. Farley and N. Ye. "QoS Requirements of Network Applications on the Internet, April 2004.
- [18] Youjian Zhao and Tielei Zhang and Yong Cui. "Cotconnectionless Quality-of-Service Routing Framework". *IEEE CESA*, October 2006.
- [19] Z. Wang and J. Crowcroft. "QoS Routing for Supporting Resource Reservation". *IEEE JSAC*, September 1996.